Twenty Dollars

Cromemco

68000 Board Family

Instruction Manual

Cromemco®

68000 Board Family

Instruction Manual

CROMEMCO, Inc.
280 Bernardo Avenue
Mountain View, CA. 94043

Part No. 023-2016

June 1982

Utility Signals 32
M Bus Signal Definitions 33
Address and Data Signals 35
Control Signals 35
I/P Port Characteristics 36

Appendix A: HOW THE DPU WORKS - A PROGRAM EXAMPLE 39

Appendix B: LIMITED WARRANTY 45

LIST OF TABLES AND FIGURES

Table	2-1:	DPU Specifications	4
Figure	2-1:	DPU Jumper Locations	5
Figure	2-2:	Memory Configurations	7
Figure	2-3a:	WDI-II Modifications - Component Side	10
Figure	2-3b:	WDI-II Modifications - Solder Side	11
Table	3-1:	MCU Specifications	15
Figure	3-1:	MCU Jumper Location	16
Table	3-2:	Error Determination - Row	17
Table	3-3:	Error Determination - Column	18
Table	4-1:	256MSU/512MSU Specifications	19
Table	4-2:	MSU Switch Settings	20
Figure	4-1:	256MSU Switch	21
Figure	4-2:	512MSU Switch	22
Figure	4-3:	DPU, MCU, and MSU Boards Installed in a Standard Cromemco System	23
Figure	4-4:	Installing the M Bus Cable	24
Table	5-1:	S-100 BUS/DPU Signals	28
Table	5-2:	Status Signal States	32
Table	5-3:	M Bus Signals	34

Chapter 1

INTRODUCTION

The Cromemco 68000 board family is comprised of the **DPU** Dual Processor Unit, **MCU** Memory Control Unit, and **256MSU** & **512MSU** Memory Storage Units.

Based on the Motorola **MC68000** microprocessor chip, the Cromemco **DPU** introduces **32-bit** computing speed and power to the Cromemco line of computers. The DPU is a **Dual** Processor Unit, also incorporating an 8-bit **Z-80A** microprocessor chip to ensure compatibility with existing software. Microprocessor selection is under software control, allowing the programmer to select the chip of choice for any particular application or routine. An existing Z-80 program, for example, may be run on the DPU with a subroutine or two rewritten in 68000 code to speed things up.

Appendix A of this manual contains a Z-80 Assembly language program which demonstrates the use of software controlled microprocessor switching techniques.

And what about that program which doesn't quite all fit in main memory? With a 24-bit wide address bus, the MC68000 can address up to 16 Mbytes of memory. This is not bank-selected memory. The MC68000 can **directly address 16 megabytes** of RAM.

The Cromemco **MCU** Memory Controller Unit acts as an interface between the S-100/IEEE-696 bus and the Cromemco **M bus** which runs between the MCU and its associated MSUs. A single MCU works in conjunction with up to eight MSU boards providing not only an interface, but an error detection, correction, and logging system.

The Cromemco **256MSU** provides 128K 22-bit words, each of which includes six bits for error detection and correction. The **512MSU** provides 256K 22-bit words. This scheme allows all single-bit errors to be corrected and double-bit errors to be detected and logged.

The Cromemco 68000 board family **conforms to the S-100/IEEE-696** bus standard.

Chapter 2

THE DPU DUAL PROCESSOR UNIT

INTRODUCTION

The Cromemco DPU Dual Processor Unit incorporates two microprocessors on a single board. The MC68000 provides the power and speed of a new generation of 32 bit microprocessor chips while the Z-80A guarantees compatibility with most existing hardware and software. One of the two microprocessors is in charge of the bus at any given point in time, yielding control to the other as required by the software.

The MC68000 microprocessor has 32-bit wide internal registers, a 16-bit wide external data bus, and a 24-bit wide external address bus. It can directly address 16 megabytes of memory and has 56 instruction types, 5 data types, and 14 address modes yielding an instruction set comprised of over 1000 instructions. These features make the MC68000 extremely fast and versatile.

The Z-80A allows the DPU to maintain compatibility with most existing Cromemco hardware and software. The DPU can be used as a direct replacement for a ZPU board.

Table 2-1 shows the operational specifications for the DPU board.

Table 2-1: DPU SPECIFICATIONS

Processors:	MC68000 and Z-80A
Clock Rate:	MC68000 - 8MHz Z-80A - 4MHz System - 4MHz
Instruction Set:	MC68000 - over 1000 instructions in 56 main types Z-80A - 158 instructions including the 78 instructions of the 8080 processor
Power-on Jump:	Jumper selectable to any 4K memory boundary within the first 64K page. Preselected for standard Cromemco systems.
Processor Control:	Software controlled switching between MC68000 and Z-80A.
Bus:	S-100/IEEE-696
Power Requirements:	+8 volts @ 2.0 amps
Operating Environment:	0 to 55 degrees C

SETUP AND INSTALLATION

Switch Settings

There are no switches on the DPU.

Jumper Selectable Options

The Cromemco DPU board is ready to be used in your Cromemco system as it is shipped from the factory. There are three jumpers which may be used for experimental purposes. The locations of these jumpers are shown in Figure 2-1.

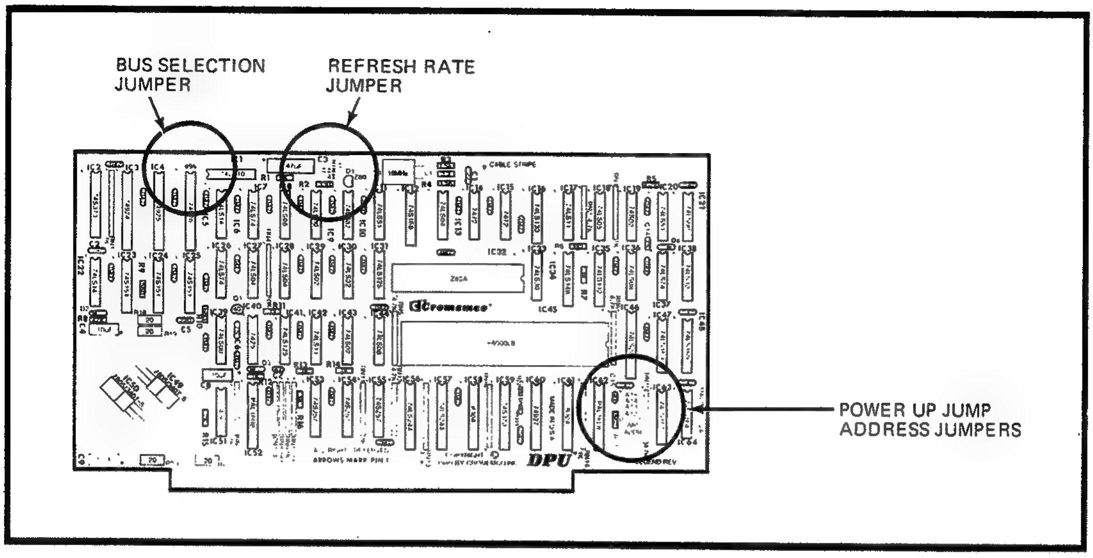

Figure 2-1: DPU JUMPER LOCATIONS

Bus Selection Jumper - This jumper is not needed when using Cromemco boards.

Memory Refresh Rote - This jumper is not needed when using Cromemco boards. As shipped from the factory, there is a trace between the common node and the X2 solder pad. During MC68000 operation, this causes a refresh cycle to be inserted approximately every 16 microseconds. Cutting this trace and installing a jumper between the common node and the X1 solder pad doubles the time between refresh cycles.

Power Up Address - This jumper is not needed in standard Cromemco systems. The power up address is factory set to C000h. This set of jumpers allows the address at which execution starts when the system is turned on or reset to be changed.

Installation

Turn off all power to the system and unplug it before installing or removing any circuit board.

The DPU is independent of all other boards and may be installed in any available bus slot. No cables are connected to the DPU.

MEMORY SUPPORT

The DPU supports both vertical and horizontal memory configurations.

Horizontal memory, arranged in banks, is used by the 8-bit Cromix Operating System. This memory configuration allows the operating system to select one of up to seven banks of 64 Kbytes of memory for a total of 448 Kbytes of addressable memory. Horizontal memory requires the use of Cromemco 64KZ memory boards and does not support the D-series Cromix Operating System. CDOS and the 8-bit Cromix Operating System are supported by horizontal memory configurations.

Vertical memory is used by the D-series Cromix Operating System. This memory configuration allows the operating system to directly address up to 16 Mbytes of memory without the use of bank selection techniques. Vertical memory requires the use of Cromemco MSU series boards together with an MCU. CDOS is not supported by vertical memory configurations. All Cromemco I/O boards are compatible with a DPU configured with vertical memory except the SDI/48KTP and the original WDI. Refer to the sections of this chapter titled **Using a DPU in a Hard Disk System** and **Using a DPU with an SDI/48KTP.**

Vertical and Horizontal memory cannot be combined within a single system. You cannot use 64KZ boards in the same system with MSU series boards.

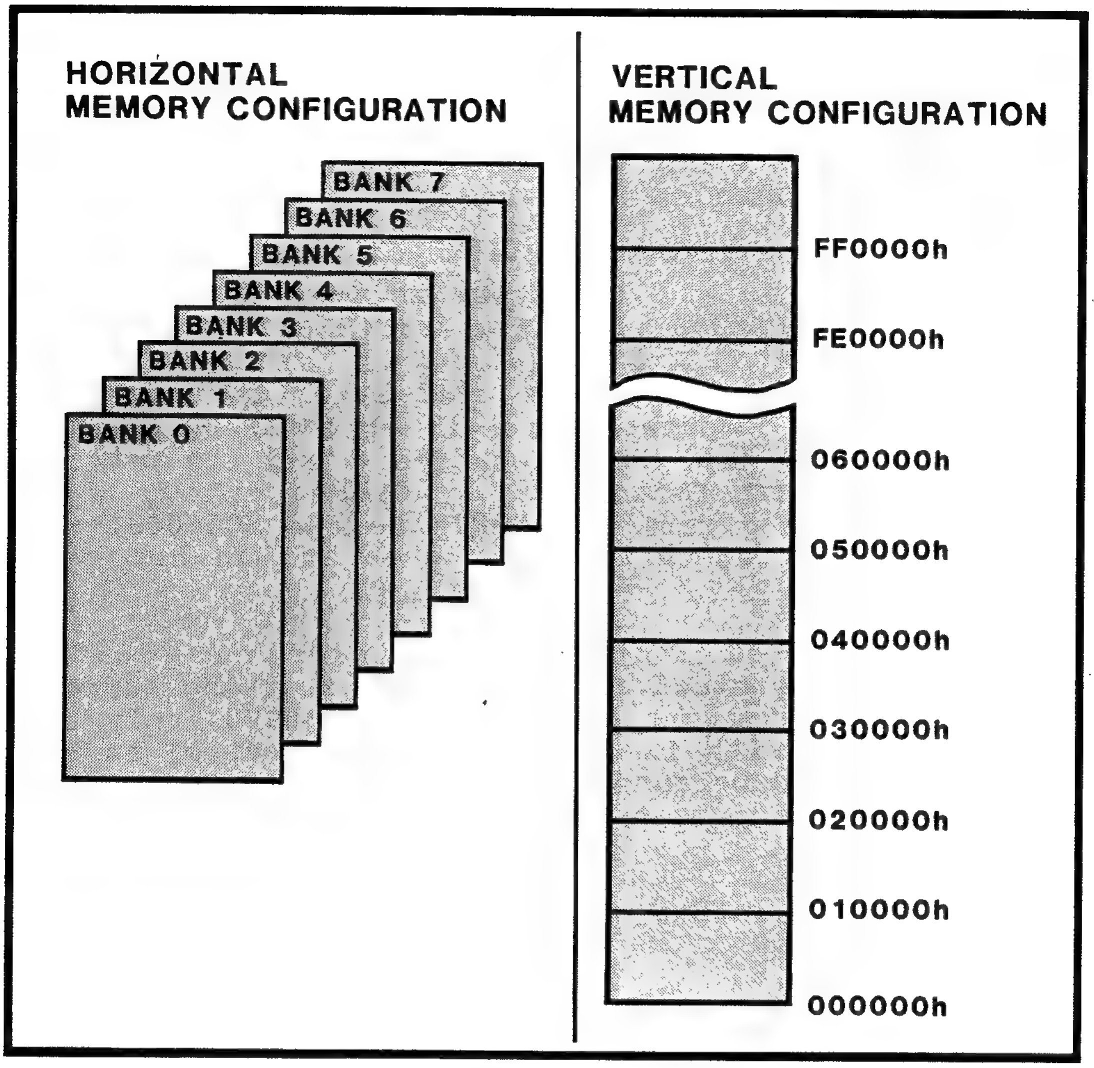

Figure 2-2: MEMORY CONFIGURATIONS

INPUT AND OUTPUT

While in the Z-80A mode of operation, the DPU performs input and output by executing Z-80 **in** and **out** instructions, specifying the desired values and ports.

While in the MC68000 mode of operation, the DPU performs input and output by reading and writing from and to the

top bytes of memory (i.e., FFFF00h - FFFFFFh). This area of memory is mapped by the DPU firmware to correspond to ports 0 - FFh.

In both modes of operation, port FFh is reserved for switching from one microprocessor to the other. Refer to the following section for an example of I/O using each microprocessor.

MICROPROCESSOR SELECTION

The DPU board automatically starts up in the Z-80A mode of operation. When a **1** is output to port FFh, the DPU switches to the MC68000 mode of operation.

The first time the MC68000 is used after the power is turned on or the computer is reset, the MC68000 obtains its stack pointer (sp) from location 0 and its program counter (pc) from location 4. Subsequently, when control is switched from the Z-80A to the MC68000, the program continues execution as though it had not been interrupted.

The following Z-80 instructions initiate MC68000 operation with the stack at 6000h and a starting address of 7000h. On subsequent calls to the MC68000, only the two instructions with the comment **switch control to the MC68000** are needed because the stack pointer and program counter have already been established.

```
        ld      hl,intdat           ; initialize sp & pc
        ld      de,0                ;    /
        ld      bc,8                ;   /
        ldir                        ;  /
        ;
        ld      a,1                 ; switch control to
        out     0ffh,a              ; MC68000
        ;
intdat: db      0, 0, 60h, 0        ; location of stack
        db      0, 0, 70h, 0        ; starting address
```

To switch from the MC68000 mode of operation to the Z-80A mode, output a **0** to port FFh. This may be done by writing to the top 64K of memory as follows.

```
    move.b     #0,0ffffffh
```

The 24-bit address to which the zero is moved may be thought of in three segments. The first **ff** indicates that the top 64K of memory is to be used. This is the area of memory which has been reserved for memory mapped I/O. The second **ff** is mandatory. The last **ff** is the port address.

USING A DPU IN A HARD DISK SYSTEM

The DPU is not compatible with the WDI hard disk interface board.

The DPU is compatible with WDI-II revision B boards, but these boards require modification for proper operation with the DPU.

WDI-II boards, revision D and above, only require proper CPU jumper selection for use with a DPU. This selection is made at the factory and, unless the boards have been modified, these boards require no further modifications or jumpers.

WARNING

Use of a WDI, an unmodified WDI-II revision B, or an improperly jumpered WDI-II revision D or higher with a DPU may result in loss of data from the hard disk. Verify proper operation of the hard disk as described in Step 17 below before proceeding.

The following modifications should only be made by your authorized Cromemco dealer or service facility. These modifications are to be made only to WDI-II revision B boards. Figures 2-3a and 2-3b show the required modifications.

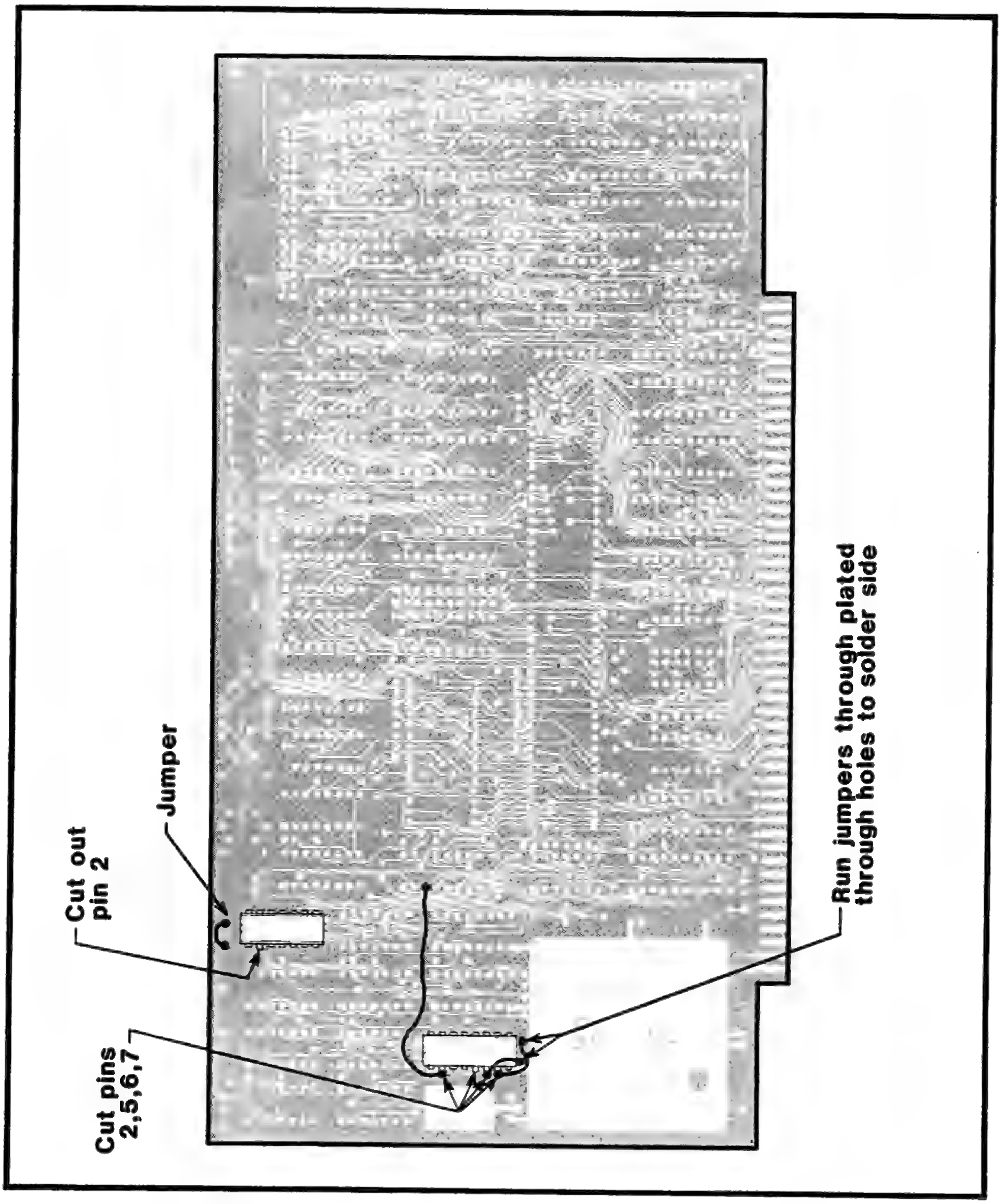

Figure 2-3a: WDI-II MODIFICATIONS - COMPONENT SIDE

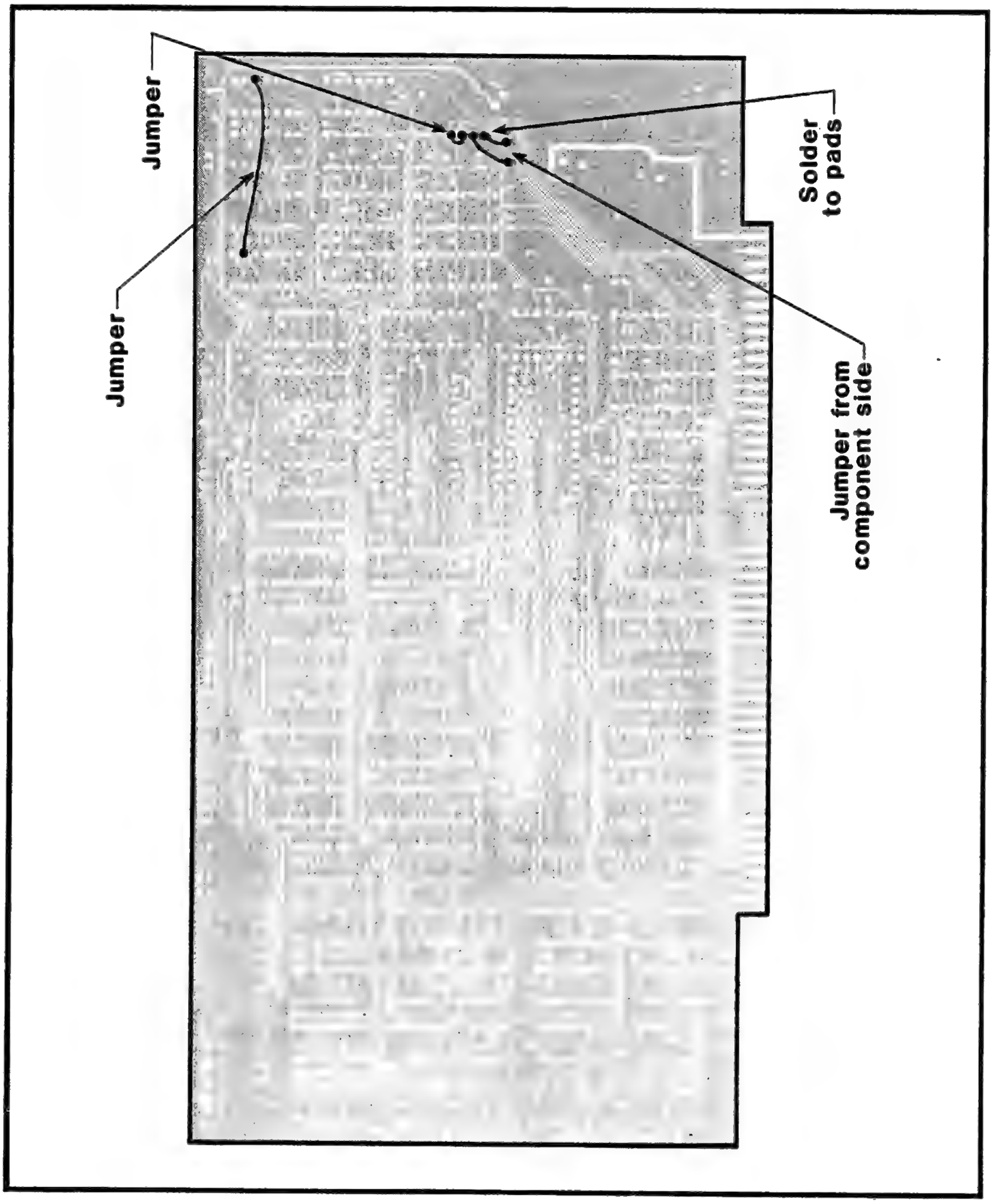

Figure 2-3b: WDI-II MODIFICATIONS - SOLDER SIDE

1. Turn off the system power and unplug the system. If the board is installed in a system, remove it.

2. Remove the nut and insulating screw which secure IC25. Bend IC25 until it is perpendicular to the board.

3. Place the board on the work area with the component side up.

4. Cut pins 2, 5, 6, and 7 of IC26, and pin 2 of IC6, as near to the board as is possible. Bend the pins out and up until they are parallel to the board surface.

5. Solder a 2-inch insulated jumper wire to pin 7 of IC26. Feed the jumper wire through the hole near the bottom right side of IC26.

6. Solder a 2-inch insulated jumper wire to pin 6 of IC26. Feed the jumper wire through the hole near the bottom left side of IC26.

7. Turn the board over so that the component side is down. Solder the jumper from pin 7 of IC26 to the board at the solder pad which was connected to pin 6 of IC26.

8. Solder the jumper from pin 6 of IC26 to the board at the solder pad which was connected to pin 7 of IC26.

9. Solder a bare jumper wire between pins 4 and 5 of IC26. Make sure that the jumper **is not shorted** to the trace that passes between pins 4 and 5.

10. Solder an insulated jumper wire between the solder pads of pin 2 of IC6 and pin 3 of IC3.

11. Turn the board over so that the component side is up. Solder an insulated jumper wire between pin 2 of IC26 and pin 3 of IC29.

12. Bend IC25 back to its original position. Secure IC25 with the nut and insulating screw which were removed in Step 2.

13. Verify that the CPU selection jumper is in the ZPU position. The jumper is at the top of IC6 in the form of a circuit trace on the circuit side of the board. If the board has been previously jumpered, it may be necessary to remove the existing jumper and install a new jumper in the ZPU position.

14. Install the WDI-II in the system. Attach the hard disk drive cable to connector J2.

15. Plug the system in and turn on the power.

16. Boot the system using a floppy disk. DO NOT access the hard disk at this time. If you are using a Cromix system, do not boot up on the hard disk and do not mount the hard disk until you have verified that it is operating properly.

17. When using the Cromix Operating System, use the Dump utility to verify proper hard disk operation. Enter the following command (with the hard disk still not mounted). Substitute **hd1** or **hd2** for **hd0** as appropriate.

    dump /dev/hd0

    When using CDOS, use the Dir utility to verify proper hard disk operation. Enter the following command. Substitute **f** or **g** for **e** as appropriate.

    dir e:

18. If the dump is normal and there are no hard disk error messages displayed, the hard disk is operating properly and it may be mounted or established as the root device.

    If error messages are displayed, check all cables and then check all modifications. Do not mount the hard disk until it can be dumped without any error messages being displayed.

USING A DPU WITH AN SDI/48KTP

The D-series Cromix Operating System and the MCU/MSU boards will not support an SDI graphics system.

The DPU may be used with CDOS or an 8-bit Cromix system incorporating 64KZ memory boards in conjunction with an SDI graphics system.

Chapter 3

THE MCU MEMORY CONTROL UNIT

INTRODUCTION

The MCU Memory Control Unit performs three functions in a DPU based system. It controls DPU access to MSU memory boards, provides the operating system with information about errors logged by the error detection circuits, and corrects single-bit errors.

A modified Hamming algorithm is used for error detection and correction. This method can detect, log, and correct single-bit errors and can detect and log double-bit errors. Error logging data indicates the MSU board on which each error occurred, the number of times the error occurred, the chip row, and, for single-bit errors, the chip column. Error logging provides an early indication of impending memory problems and the necessary information for quick service, should it be required.

Table 3-1 shows the operational specifications for the MCU board.

Table 3-1: MCU SPECIFICATIONS

Support Capacity:	Up to eight MSU memory storage boards.
Addressable Memory Locations:	16 megabytes
Buses:	Directly compatible with S-100/IEEE-696. The M bus connects MCU and MSU boards.
Power Requirements:	+8 volts @ 1.5 amps
Operating Environment:	0 to 55 degrees C

SETUP AND INSTALLATION

Switch Settings

There are no switches on the MCU.

Jumper Selectable Options

The Cromemco MCU board is ready to be used in your Cromemco system as it is shipped from the factory. There is one set of jumpers which may be used to change the I/O address for experimental purposes. The location of this jumper is shown in Figure 3-1.

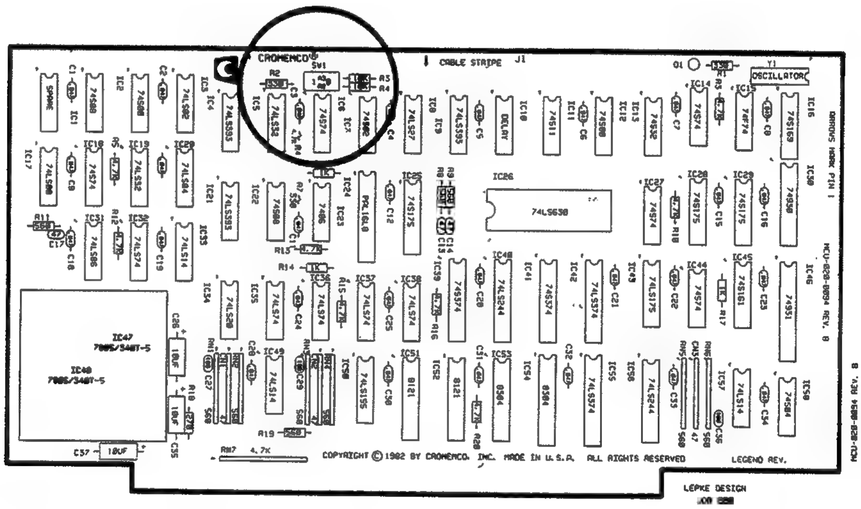

Figure 3-1: MCU JUMPER LOCATION

Installation

Please refer to Chapter 4 for installation information.

ERROR DETECTION AND LOGGING

Please refer to Chapter 5, **I/O Port Characteristics** while reading this section.

An error is reported by bit D7 of the Control Read byte and the red LED on the MCU board. After the error has been logged by the operating system, this bit may be cleared and the light turned off by the Clear Error command. Error correction will continue regardless of the state of this bit.

Determining the chip which generated the error is a three step process. The MSU board, the row, and the column containing the chip must be identified.

The **MSU ID** byte identifies the board on which the error occurred. Convert the MSU ID byte to binary and match, from left to right, the switch settings on the MSU boards. The 512MSUs have only five switch bits to match while the 256MSUs have six. An MSU ID which contains zeroes in the five (512MSU) or six (256MSU) most significant binary positions always indicates the MSU board addressed at zero. Refer to Table 4-2 for a list of binary values of switches for boards at different memory locations.

The **Chip Row ID** bits identify the row in which the error occurred. Refer to Table 3-2.

Table 3-2: MEMORY ERROR LOCATION BY CHIP ROW

Chip Row ID	Status Bits D5*	Status Bits D4
0	0	0
1	0	1
2	1	0
3	1	1

*Ignore D5 and rows 2 and 3 when MSU ID byte indicates error on 256MSU.

The **Syndrome Code Data** bits identify the column in which the error occurred. Refer to Table 3-3. The Syndrome Code is only valid for single-bit errors.

Table 3-3: ERROR DETERMINATION - COLUMN

Chip Column	Syndrome Code Bits					
	D5	D4	D3	D2	D1	D0
00	1	1	0	1	0	0
01	1	1	0	0	1	0
02	1	1	0	0	0	1
03	1	0	1	1	0	0
04	1	0	1	0	1	0
05	1	0	1	0	0	1
06	1	0	0	1	0	1
07	1	0	0	0	1	1
08	0	1	1	1	0	0
09	0	1	1	0	1	0
10	0	1	0	1	1	0
11	0	1	0	1	0	1
12	0	1	0	0	1	1
13	0	0	1	1	1	0
14	0	0	1	1	0	1
15	0	0	1	0	1	1
16	1	1	1	1	1	0
17	1	1	1	1	0	1
18	1	1	1	0	1	1
19	1	1	0	1	1	1
20	1	0	1	1	1	1
21	0	1	1	1	1	1

Chapter 4

THE MSU MEMORY STORAGE UNIT

INTRODUCTION

Cromemco manufactures two types of MSU boards, each with a different storage capacity. The 256MSU board has a storage capacity of 256 Kbytes of RAM while the 512MSU has 512 Kbytes. These boards are realized as two (256MSU) and four (512MSU) rows of twenty-two 64 Kbit dynamic RAM chips. Each 16-bit data word has six additional bits associated with it. These bits are used by the MCU for error detection and correction.

Table 4-1: 256MSU/512MSU SPECIFICATIONS

Memory Capacity:	256MSU - 128K words of 22 bits (including 6 error detection bits) 256K bytes 512MSU - 256K words of 22 bits (including 6 error detection bits) 512K bytes
Memory Type:	64K RAM 150 nanosecond access time
Buses:	Directly compatible with S-100/IEEE-696. The M bus connects MCU and MSU boards.
Power Requirements:	+8 volts @ 1.5 amps
Operating Environment:	0 to 55 degrees C

SETUP AND INSTALLATION

Switch Settings

All switches on either a single 256MSU or 512MSU board in a system should be set to zero (off). This will address the board at zero. If more than one MSU board is used, the switches should be set according to Table 4-2. Do not address memory boards so that they overlap.

Table 4-2: MSU SWITCH SETTINGS

Lower Memory Boundary		256MSU Switch Bit	512MSU Switch Bit
hex (0000)	decimal (K)	1 2 3 4 5 6	1 2 3 4 5
00	0	0 0 0 0 0 0	0 0 0 0 0
04	256	0 0 0 0 0 1	
08	512	0 0 0 0 1 0	0 0 0 0 1
0C	768	0 0 0 0 1 1	
10	1024	0 0 0 1 0 0	0 0 0 1 0
14	1280	0 0 0 1 0 1	
18	1536	0 0 0 1 1 0	0 0 0 1 1
1C	1792	0 0 0 1 1 1	
20	2048	0 0 1 0 0 0	0 0 1 0 0
.	.	.	.
.	.	.	.
.	.	.	.
F0	15360	1 1 1 1 0 0	1 1 1 1 0
F4	15616	1 1 1 1 0 1	
F8	15872	1 1 1 1 1 0	1 1 1 1 1 *
FC	16128	1 1 1 1 1 1 *	

*reserved for memory mapped I/O

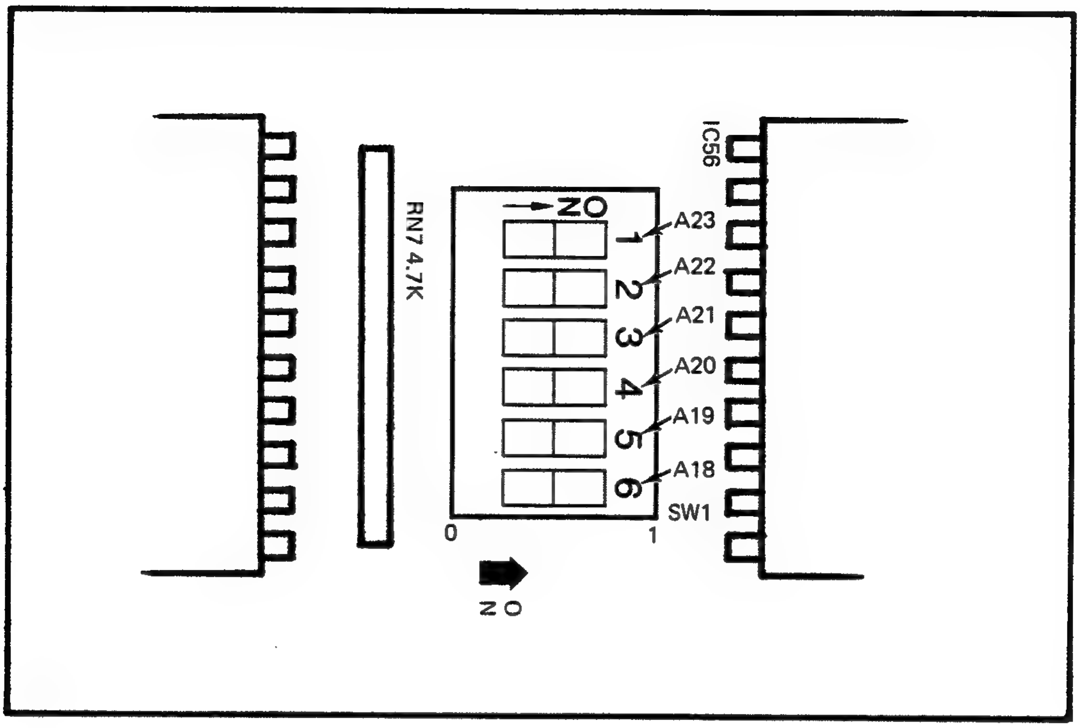

Figure 4-1: 256MSU SWITCH

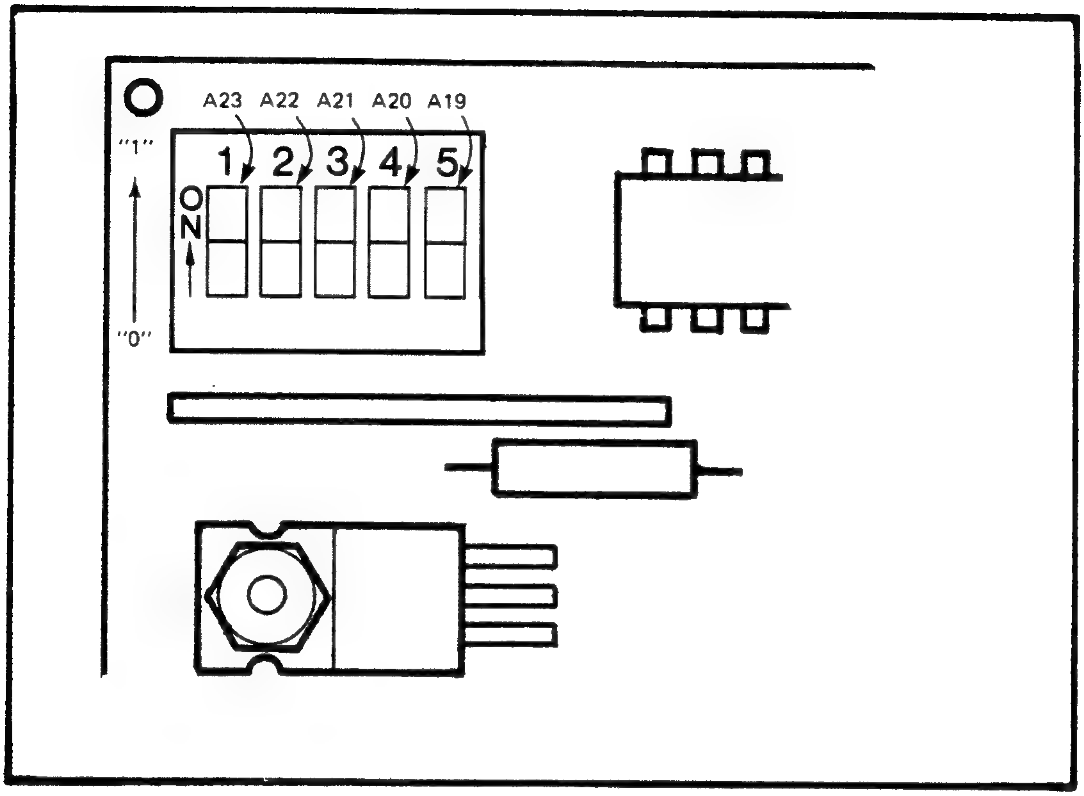

Figure 4-2: 512MSU SWITCH

Jumper Selectable Options

There are no jumper selectable options on either MSU board.

Installation

The MCU and MSU boards share a common cable and therefore must be installed in a physical group. The order or spacing is not important except for the positioning of the overhead M Bus cable.

These boards may be installed as shown in Figure 3-2. When all MCU and MSU boards are installed, take the M bus cable and press it firmly onto the boards as shown in Figure 3-3.

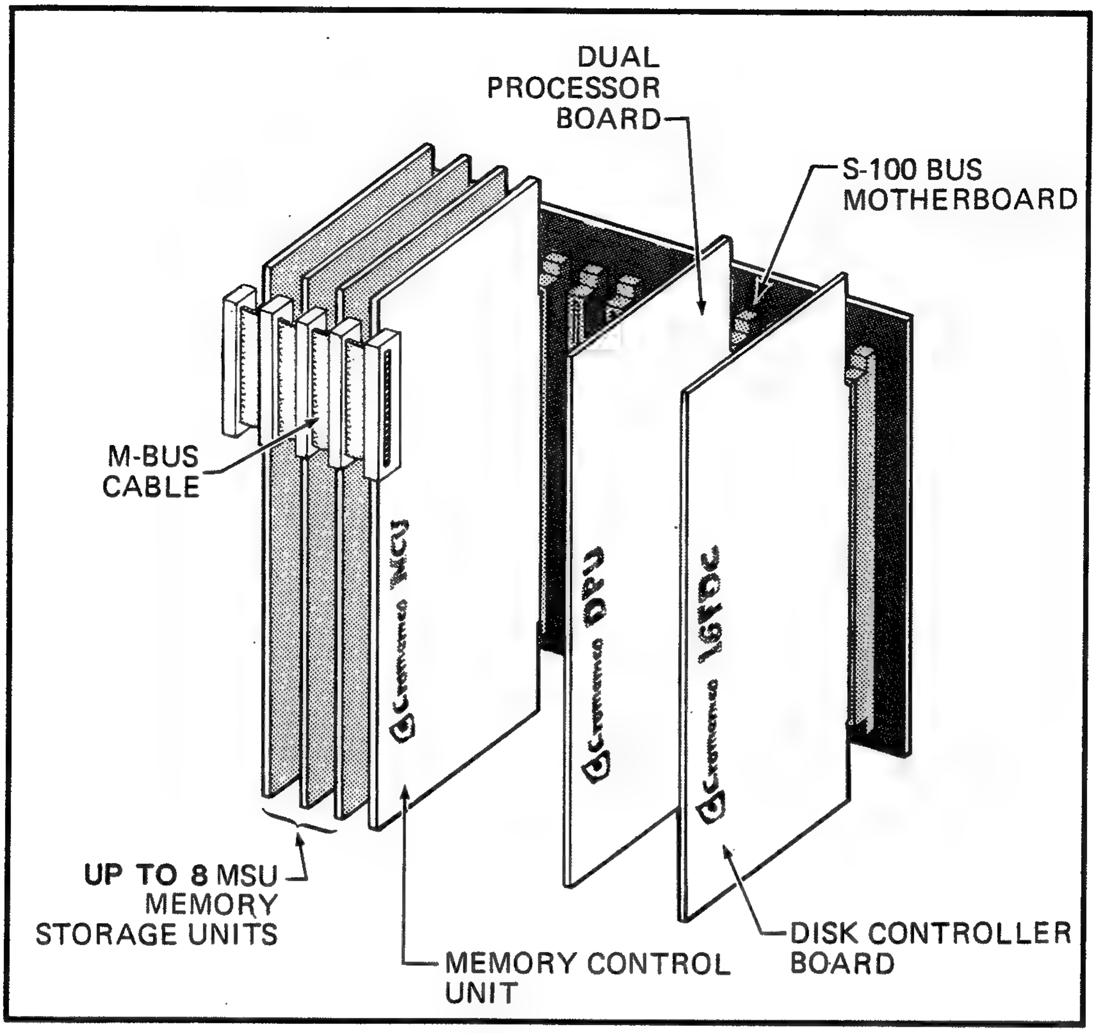

Figure 4-3: DPU, MCU, AND MSU BOARDS INSTALLED IN A STANDARD CROMEMCO SYSTEM

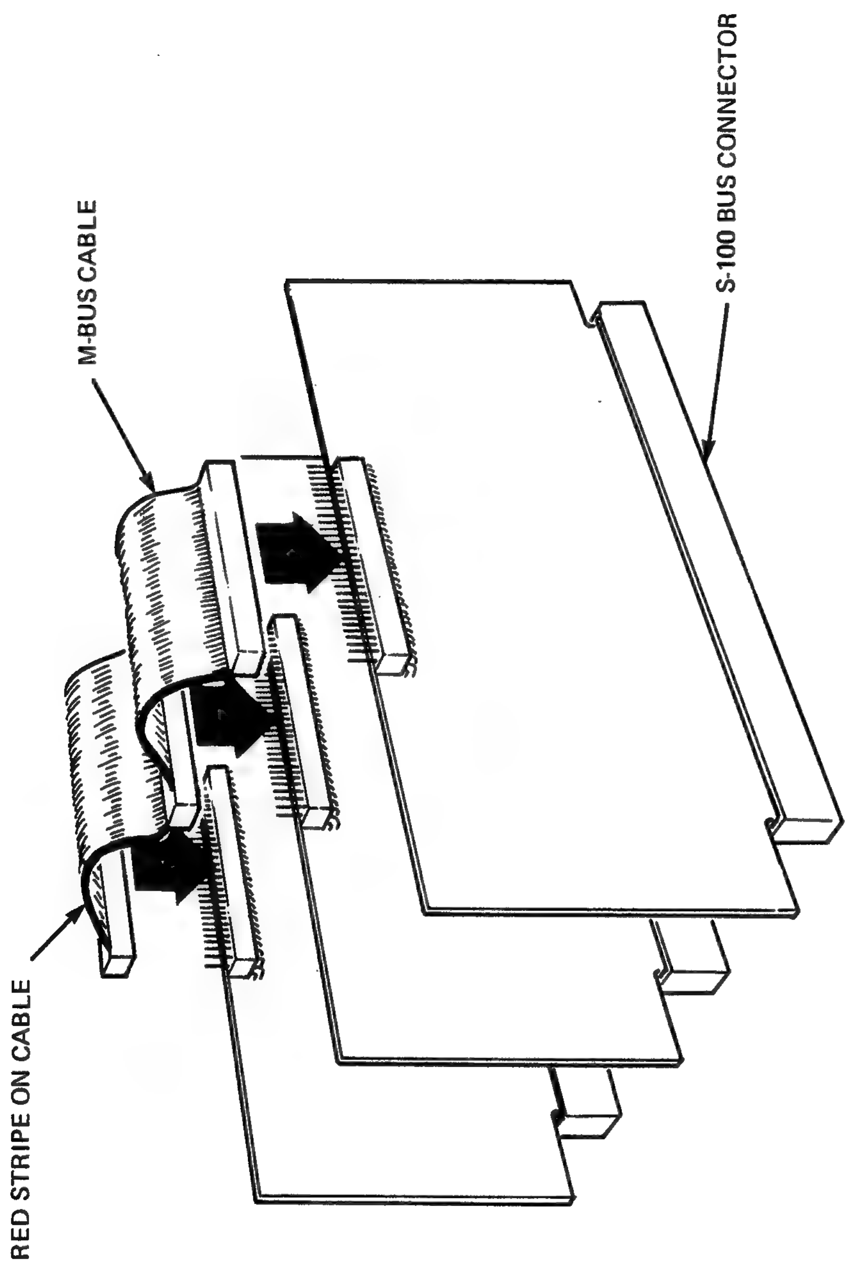

Figure 4-4: INSTALLING THE M BUS CABLE

Up to eight MSU boards can be used with a single MCU. Cromemco supplies a standard cable with each MCU board for use in systems with one or two MSU boards. For systems with more than two MSU boards, use one of the cables listed below.

M Bus Cable Description	Cromemco Part Number
connects MCU and 1-2 MSUs	519-0150 (standard)
connects MCU and 3-4 MSUs	519-0149
connects MCU and 5-8 MSUs	519-0148

When more than two MSU boards are used in a system, use only the proper cable as specified above.

Chapter 5

BUS SIGNALS AND I/O PORT CHARACTERISTICS

S-100 BUS SIGNALS

The introduction of a new CPU always causes some S-100 bus signals to be redefined, some to be deleted, and others to be added. The Z-80A, used in the new DPU central processor, was the heart of the last generation of Cromemco computers. Therefore, the Z-80A, has influenced signal redefinition very little. Major changes have occurred because the DPU uses two microprocessors, one of which is the new MC68000. Most of the new signals concern either board level switching between the two processors, the 16-bit data bus, the 24-bit address bus, or control differences.

The signals of Cromemco's S-100 bus can be grouped into seven functional categories as follows.

1. address
2. control inputs
3. control outputs
4. data
5. control of DMA data transfers
6. status
7. utility

Table 5-1 shows each S-100 bus connection, signal mnemonic, name, and function. A bus signal mnemonic not only indicates the signal purpose, but also the active state. Knowing the active state will reveal the logic, binary, and electrical states that are to be expected on a given signal line. When the mnemonic is followed by an **asterisk**, the signal is **active low**. Without the asterisk, the signal is active high. For example, if the signal is GAPK* and the electrical state is low (a more negative voltage), the logic state is true and the binary state is 1. For an active high signal, such as GAPK, an electrical high (more positive voltage) is a logical true and a binary state of 1.

Table 5-1: S-100 BUS/DPU SIGNALS

Pin Number	Mnemonic	Signal Name	Function	Pin Number	Mnemonic	Signal Name	Function
1	+8V		U	51	+8V		P
2	+18V		U	52	-18Vur		P
3	XRDY	External Ready	CI	53		undefined	
4		undefined		54	SLVCLR*	Slave Clear	U
5		undefined		55		undefined	
6		undefined		56		undefined	
7		undefined		57		undefined	
8		undefined		58	sXTRQ*	Sixteen Request	S
9		undefined		59	A19	Extended Address Bit 19	A
10		undefined		60	SIXTN*	Sixteen Acknowledge	CI
11		undefined		61	A20	Extended Address Bit 20	A
12	NMI*	Non-maskable Interrupt	CI	62	A21	Extended Address Bit 21	A
13	PWRFAIL*	Power Fail	U	63	A22	Extended Address Bit 22	A
14		undefined		64	A23	Extended Address Bit 23	A
15	A18	Extended Address Bit 18	A	65	MREQ*	Memory Request	U
16	A16	Extended Address Bit 16	A	66	RFSH*	Refresh (Memory)	U
17	A17	Extended Address Bit 17	A	67	MEMDSB*	Memory Disable	U
18	SDSB*	Status Disable	DMA	68	MWRT	Memory Write	S
19	CDSB*	Control Output Disable	DMA	69		undefined	
20		Ground	U	70		Ground	U
21		undefined		71	Z80/68*	Z80/MC68000 Processor Active	CO
22	ADSB*	Address Disable	DMA	72	pRDY	Ready	CI
23	DODSB*	Data Out Disable	DMA	73	pINT*	Interrupt Request	CI
24	ϕ2	System Clock	U	74	pHOLD*	Hold	CI
25	pSTVAL*	Status Valid Strobe	CO	75	RESET*	Reset (Bus Masters)	U
26	pHLDA	Hold Acknowledge	CO	76	pSYNC	Synchronize	CO
27	EXTAD*	Extended Address Enable	U	77	pWR*	Write	CO
28		undefined		78	pDBIN	Data Bus In	CO
29	A5	Address Bit 5	A	79	A0	Address Bit 0	A
30	A4	Address Bit 4	A	80	A1	Address Bit 1	A
31	A3	Address Bit 3	A	81	A2	Address Bit 2	A
32	A15	Address Bit 15	A	82	A6	Address Bit 6	A
33	A12	Address Bit 12	A	83	A7	Address Bit 7	A
34	A9	Address Bit 9	A	84	A8	Address Bit 8	A
35	DO1	Data Out Bit 1/Data Bit 1	D	85	A13	Address Bit 13	A
36	DO0	Data Out Bit 0/Data Bit 0	D	86	A14	Address Bit 14	A
37	A10	Address Bit 10	A	87	A11	Address Bit 11	A
38	DO4	Data Out Bit 4/Data Bit 4	D	88	DO2	Data Out Bit 2/Data Bit 2	D
39	DO5	Data Out Bit 5/Data Bit 5	D	89	DO3	Data Out Bit 3/Data Bit 3	D
40	DO6	Data Out Bit 6/Data Bit 6	D	90	DO7	Data Out Bit 7/Data Bit 7	D
41	DI2	Data In Bit 2/Data Bit 10	D	91	DI4	Data In Bit 4/Data Bit 12	D
42	DI3	Data In Bit 3/Data Bit 11	D	92	DI5	Data In Bit 5/Data Bit 13	D
43	DI7	Data In Bit 7/Data Bit 15	D	93	DI6	Data In Bit 6/Data Bit 14	D
44	sM1	Instruction Fetch Cycle	S	94	DI1	Data In Bit 1/Data Bit 9	D
45	sOUT	Data Output Cycle	S	95	DI0	Data In Bit 0/Data Bit 8	D
46	sINP	Data Input Cycle	S	96	sINTA	Interrupt Acknowledge	S
47	sMEMR	Memory Read Cycle	S	97	sWO	Write Cycle	S
48	sHLTA	Halt Acknowledge	S	98	ERROR*	Error	U
49	CLOCK	Clock, 2 MHz	U	99	POC*	Power On Clear	U
50		Ground	U	100		Ground	U

Signal Function Categories: A=address, CI=control inputs, CO=control outputs, D=data, DMA=control for DMA data transfers, S=status, and U=utility.

S-100 BUS/DPU SIGNAL DEFINITIONS

The following list provides detailed definitions of each S-100 bus signal. Additional information about signal interaction with other boards may be obtained by consulting the appropriate board reference manual.

Address Bus Signals

The address bus, consisting of parallel signal paths of 24 bits, selects specific memory locations or I/O ports. The MC68000 drives all 24 bits directly. During Z-80A operation, the Z-80A drives the lower 16 bits and an auxiliary latch drives the upper 8. On power up or reset, the auxiliary latch is reset.

Memory Addressing - Memory addressing uses 24 bits, **A0 through A23**, providing a 16 Mbyte address range.

I/O Addressing - I/O addressing uses 8 bits, **A0 through A7**, allowing 1 of 256 ports to be selected.

Control Input Signals

CPU control input signals are output by bus slaves such as peripheral interface boards and memory boards. These signals synchronize the operation of the CPU and a bus slave.

External Ready - The **XRDY** signal synchronizes the response of a bus master to a bus slave. It can start and stop CPU operation.

Hold or Bus Request - The **pHOLD*** signal is a DMA bus request generated by the DMA controller of a bus slave. The CPU determines when the request shall be granted and, at the appropriate time, outputs the pHLDA signal.

Interrupt Request - The **pINT*** signal is initiated by a bus slave. Each board may output an interrupt request on this signal line connected to the DPU. The interrupts are serviced according to priorities established by the software and/or hardware.

Non-Maskable Interrupt Request - In a manner similar to pINT*, the **NMI*** signal is initiated by a bus slave. The difference between the signals is that NMI* cannot be masked by software and must be serviced by the DPU as soon as possible.

Ready - The **pRDY** signal is active during normal microprocessor operation. The negation of pRDY is the wait-state request, WAIT*, indicating that an addressed memory board or peripheral is not ready for data transfer.

Sixteen Acknowledge - The **SIXTN*** signal indicates that a request for 16-bit data transfer has been granted and that the transfer may begin. Refer also to the sXTRQ* status signal.

Control Output Signals

The control output signals are output by the DPU to control data transfer and provide the required timing reference.

Data Bus Input Strobe - The **pDBIN** read signal strobes data from the addressed slave onto the data bus.

Hold Acknowledge - The **pHLDA** signal indicates to the board with the highest priority bus request that the DPU has relinquished control of the system bus and that DMA controlled data transfer may begin. Refer to the DMA control signals and pHOLD*.

Status Valid Strobe - The **pSTVAL*** signal indicates that the address and status signals present on the bus are stable and valid. Refer to pSYNC and the status signals.

Synchronization - The **pSYNC** signal indicates the start of a new bus cycle. Refer to pSTVAL* and the status signals.

Write - The **pWR*** write signal strobes data from the data bus to the addressed slave.

Data Bus Signals

The data bus in realized as 16 parallel data lines.

Under the control of the Z-80A, the bus is used as two unidirectional 8-line data buses. The data input lines, **DI0 through DI7**, bring data to the DPU while the data output lines, **DO0 through DO7**, transfer data from the DPU.

Under the control of the MC68000, all 16 lines, **D0 through D15**, are used as a bidirectional data bus. Refer also to sXTRQ*, SIXTN*, and the control output signals.

DMA Control Signals

The pHOLD* signal is issued by a board when it requires control of the bus for DMA. The DPU acknowledges pHOLD* with a pHLDA signal when it is ready to relinquish control of the bus. At the same time, the following signals disable the CPU bus buffers, effectively isolating the CPU from the bus and allowing the DMA controller to become the bus master.

1. Address Disable **ADSB***
2. Control Output Disable **CDSB***
3. Data Output Disable **DODSB***
4. Status Disable **SDSB***

Status Signals

The following status signals identify the bus cycle in progress and indicate the purpose of the address currently on the bus.

1. Memory read **sMEMR**
2. Operating Instruction Code Fetch **sM1**
3. Input **sINP**
4. Output **sOUT**
5 Memory Write **MWRT**
6. Write cycle **sWO***
7. Interrupt Acknowledge **sINTA**
8. Halt Acknowledge **sHLTA**
9. Sixteen bit data bus request **sXTRQ***

The status signals are best defined in terms of the bus cycle each represents. Table 5-2 shows the electrical state of the status signal for each bus cycle.

Status Signals	Bus Cycle Type						
	Memory Read	Op Code Fetch	Memory Write	Output	Input	Interrupt Acknowledge	Halt Acknowledge
sMEMR	H	H	L	L	L	L	X
SM1	L	H	L	L	L	X	X
sWO*	H	H	L	L	H	H	H
sOUT	L	L	L	H	L	L	L
MWRT	L	L	H	L	L	L	L
sINP	L	L	L	L	H	L	L
sINTA	L	L	L	L	L	H	L
sHLTA	L	L	L	L	L	L	H
sXTRQ*(8-bit)	H	H	H	H	H	H	X
sXTRQ*(16-bit)	L	L	L	L	L	L	X

Table 5-2: STATUS SIGNAL STATES

Utility Signals

Utility signals are necessary to the overall operation of the system. They include power supply and power supply status signals, the system clock, and generalized reset and error signals.

+8 Volts Unregulated - This is the only supply voltage required for the DPU, MCU, and MSU boards. The +18 and -18 Volts Unregulated lines are available for other boards. Regulation of the supply voltage is performed on each board.

Grounds - Signal grounds are connected together on the S-100/IEEE-696 bus.

Phase 2 System Clock - The DPU generates the $\phi 2$ 4 MHz system clock signal.

Clock - The 2 MHz **CLOCK** signal is independent of all other bus timing signals. It may be used by circuits to generate time periods or other functions requiring a fixed input frequency.

Reset - The **RESET*** signal resets all bus masters and slaves including the DPU.

Slave Clear - The **SLAVE CLR*** signal resets those bus slaves which monitor this signal.

Power-On Clear - The **POC*** signal is active only at the time that power is applied to the system. It issues the RESET* and SLAVE CLR* signals.

Memory Disable - The **MEMDSB*** signal inhibits operation of those memory boards capable of responding to this signal. Operation of Cromemco 64KZ boards will be inhibited by this signal while MSU boards will not be affected.

Memory Request - The **MREQ*** signal indicates that the address on the bus is valid for a memory read or write function.

Refresh - The **RFSH*** signal indicates when the Z-80A is performing a refresh cycle.

Z-80A/MC68000 Processor Active - When high, this signal indicates that the Z-80A microprocessor is controlling the bus. When low, it indicates that the MC68000 is controlling the bus.

M BUS SIGNAL DEFINITIONS

The **M** bus, along with the S-100 bus, provides parallel connections between the MCU and MSU boards. Control, refresh addresses, data, and utility signals are sent from the MCU to the MSU boards. Status and data signals are returned to the MCU by the MSU boards. Table 5-3 lists each **M** bus signal and indicates its function.

Table 5-3: M BUS SIGNALS

Pin Number	Mnemonic	Signal Name	Function
1	GND	Ground	U
2	RFSHA3	Refresh Address Bit 3	A, U
3	RFSHA7	Refresh Address Bit 7	A, U
4	RFSHA2	Refresh Address Bit 2	A, U
5	RFSHA6	Refresh Address Bit 6	A, U
6	RFSHA1	Refresh Address Bit 1	A, U
7	RFSHA5	Refresh Address Bit 5	A, U
8	RFSHA0	Refresh Address Bit 0	A, U
9	RFSHA4	Refresh Address Bit 4	A, U
10	RESET*	Reset	U
11	RD0	Ram Data Bit 0	D
12	RD1	Ram Data Bit 1	D
13	RD2	Ram Data Bit 2	D
14	RD3	Ram Data Bit 3	D
15	RD4	Ram Data Bit 4	D
16	RD5	Ram Data Bit 5	D
17	RD6	Ram Data Bit 6	D
18	RD7	Ram Data Bit 7	D
19	RD15	Ram Data Bit 15	D
20	RD8	Ram Data Bit 8	D
21	RD14	Ram Data Bit 14	D
22	RD9	Ram Data Bit 9	D
23	RD13	Ram Data Bit 13	D
24	RD10	Ram Data Bit 10	D
25	RD12	Ram Data Bit 12	D
26	RD11	Ram Data Bit 11	D
27	MSU SEL*	MSU Select	S
28	+5	+5 Volts	C
29	RFSH	Refresh	U
30	+5	+5 Volts	C
31	COLUMN*	Column Address Enable	C
32	GND	Ground	U
33	ROW*	Row Address Enable	C
34	GND	Ground	U
35	ENRAS*	Enable Row Address Strobe	C
36	GND	Ground	U
37	CAS	Column Address Strobe	C
38	GND	Ground	U
39	ENRAM DATA*	Enable Ram Data	C
40	GND	Ground	U
41	WRITE/(READ*)	Write/Read Enable	C
42	GND	Ground	U
43	CHECK BIT1	Check Bit 1	D
44	CHECK BIT4	Check Bit 4	D
45	CHECK BIT3	Check Bit 3	D
46	CHECK BIT5	Check Bit 5	D
47	CHECK BIT2	Check Bit 2	D
48	CHECK BIT0	Check Bit 0	D
49	EN DIAG DATA*	Enable Diagnostic Data	U
50	GND	Ground	U

Signal Function Categories: A=address, C=control, D=data, S=status, and U=utility

The following list provides detailed definitions of each **M** bus signal.

Address and Data Signals

Refresh Address Lines - The MCU places the refresh address on the **RFSHA0** through **RFSHA7** lines. Refer to the RFSH signal.

RAM Data - The **RD0 through** RD15 data lines are bidirectional. Data is transferred between the S-100 bus and MSU boards via the MCU board and the M bus. Refer to the EN DIAG DATA* and EN RAM DATA* signals.

Check Bit Data - The data on the **CHECK BIT0 through CHECK BIT5** data lines is calculated from a 16-bit data word being written to memory. The data on these lines is written to the six error detection bits associated with each word in memory. During the memory read cycle, this data is compared to the check bits calculated from the retrieved data word to determine if an error occurred.

Control Signals

Reset - The **RESET*** signal is derived from the system reset by the MCU. It is not used.

MSU Select - The **MSU SEL*** status signal is output by the selected MSU board, acknowledging that it has been selected for data transfer.

Refresh - The **RFSH** control signal is output by the MCU during a refresh cycle. It causes the refresh address to be placed on the address inputs of the RAM chips and the Row Address Strobes to be issued to all RAM chips on all MSUs simultaneously when the EN RAS* strobe is received.

Column Address Enable - The **COLUMN*** signal enables the buffers for address lines A10 through A17 on the 256MSU and A11 through A18 on the 512MSU. This address is latched into the column address inputs of each memory chip. The column address, along with the row address, selects 1 bit of 64 Kbits.

Row Address Enable - The **ROW*** signal enables the buffers for address lines A2 through A9 on the 256MSU and A3 through A10 on the 512MSU. This address is latched into the row address inputs of each memory chip. The row address, along with the column address, selects 1 bit of 64 Kbits.

Enable Row Address Strobe - The **EN RAS*** strobe causes the chip row address strobe (RAS) to be output to a row of memory chips. Address bit A1 selects 1 row of 2 to be strobed on the 256MSU. Address bits A1 and A2 select 1 row of 4 to be strobed on the 512MSU. In the refresh mode, RFSH selects all chip rows and EN RAS* causes them to be strobed simultaneously. Strobing RAS latches the row address into the RAM chip.

Column Address Strobe - The **CAS** signal asserts the column address strobe of each RAM chip.

Enable RAM Data - The **EN RAM DATA*** signal enables the RAM data bus buffers on the MSU involved in the memory operation.

Write - The **WRITE** signal causes the individual write signals to be issued to the memory chips.

Enable Diagnostic Data - The **EN DIAG DATA*** signal inhibits the RD0 through RD7 buffers and enables the diagnostic ROM.

I/O PORT CHARACTERISTICS

The following is a list of I/O port addresses and their functions.

4Ch **Out: Control**

The **Control Write** byte sends control data from the CPU to the MCU.

D7-D2 - These bits must be zero.

D1 Enable Error Detection and Correction (EN EDAC) - This bit enables the error detection and correction circuits on the MCU board.

D0 Enable Diagnostic (EN DIAG) - This bit enables the diagnostic function of the memory system.

4Ch In: Status

The **CONTROL READ** byte sends status information from the MCU to the CPU.

D7 Error - Indicates that an error has been detected. When this bit is on, the red LED on the MCU will be lit.

D6 Unused - This bit will read as zero.

D5-D4 Chip Row ID - These signals identify the row containing the chip which generated the error. The 256MSU, having only two rows of chips, uses only D4 to identify the row while the 512MSU uses both D5 and D4. Refer to Chapter 3, **Error Logging**.

D3-D2 - not used

D1 Error Circuit Status - This signal indicates that the error detection and correction circuits are enabled.

D0 Diagnostic Circuit Status - This signal indicates that the diagnostic circuits are enabled.

4Dh In: Syndrome Code

The **Syndrome Read** byte contains information pertinent to an error condition.

D7 Diagnostic Circuit Status at Error - This bit indicates the diagnostic circuits were enabled when the error occurred. A logical false indicates that the error occurred under normal operation.

D6 Error Status - This bit indicates that the error was a single error; a logical false indicates a double error.

D5-D0 Syndrome Code Data - These data bits identify the chip column in which the error occurred. Refer to Chapter 3, **Error Logging**.

4Dh Out: Clear Error

The **Clear Error** command clears bit D7 of the Control Read status word. This byte should be set to all zeroes.

4Eh **In:** MSU ID

The **Error Address READ** byte identifies the MSU board containing the memory chip in which an error occurred. Refer to Chapter 3, **Error Logging.**

4Eh **Out:** not used

4Fh **In:** not used

4Fh **Out:** not used

FFh **Out: MC68000/Z-80A Switch**

The **MC68000/Z-80A Switch Output Byte** enables either the MC68000 or the Z-80A microprocessor on the DPU. When one is enabled, the other is disabled. Refer to Chapter 2, **Microprocessor Selection.**

D7-D1 - not used

D0 Microprocessor Selection Switch - When set, this bit turns the MC68000 on. When reset, it turns the Z-80A on.

Appendix A

HOW THE DPU WORKS - A PROGRAM EXAMPLE

The purpose of this example is to illustrate an assembly language program, written in Z-80 assembler, which calls a 68000 multiply subroutine. This program illustrates how the DPU can, by means of software instructions, switch from Z-80 operation to 68000 operation and back again.

The DPU always comes up from power-off or reset in Z-80 mode. To switch to the 68000, all that is required is an output to port 0FFh. The 68000 uses memory mapped I/O, so an output is performed by moving data to the highest page of memory. Therefore the switch back to the Z-80A is done by moving a 0 to memory location 0FFFFh (Port 0FFh).

When using the DPU with the Cromix Operating System, details of Z-80/68000 switching are handled automatically and are transparent to the user.

Cromemco 68000 Board Family
A. How the DPU Works - A Program Example

```
;
;This program prompts for two hex numbers to be input from the
;CRT.  The numbers are converted to binary and placed in memory
;for processing by the 68000 microprocessor on the DPU board.
;The 68000 executes a program that gets the two arguments and
;multiplies them. The 4 byte result is loaded back into memory,
;and control is returned to the Z-80.  The Z-80 then converts
;the 32 bit binary result into 8 hex digits and prints the results
;on the CRT. This example runs under CDOS and uses CDOS calls for
;simplicity and since it would be unwise to change processors
;yourself in a multi-user environment.
;
arg1:   equ     4000h
arg2:   equ     4100h
stack:  equ     4000h
mcnt:   equ     100d
;
        ext     binh2,ahex,prnbf$,gtlnl$        ;asmlib routines
;
start:  ld      hl,6500h        ;setup for 68000 to out 0 to 0FFh.
        xor     a
        ld      (hl),a
        ld      c,9             ;print begin message.
        ld      de,msg
        call    5               ;call CDOS
        ld      c,1             ;do read call to allow for pause.
        call    5               ;call CDOS
        ld      de,6000h        ;load a prog. for the 68000 to run.
        ld      hl,prog68       ;prog68 is the 68000 program
        ld      bc,count
        ldir                    ;move 68000 program to execution area
;
;Input two numbers to be passed to the 68000 for multiplication.
;
mainlp: ld      b,5
go_on:  push    bc
        ld      de,arg1ms
        call    prnbf$
        ld      b,4             ;set up to input four hex digits.
        ld      hl,stor1        ;point to storage for input.
        ld      d,'0'
        ld      e,'F'
        call    gTLNl$
        ld      a,(hl)          ;get number of chars. to input.
        ld      de,0
        ld      e,a             ;load this number into de,
        add     hl,de           ;and add to hl so we point to
        inc     hl              ;end of input.
        ld      a,'H'
        ld      (hl),a          ;set up string for ahex.
        ld      de,arg2ms
        call    prnbf$
        ld      b,4             ;set up to input four hex digits.
```

```
        ld      hl,stor2        ;point to storage for input.
        ld      d,'0'
        ld      e,'F'
        call    gtlnl$
        ld      a,(hl)          ;get number of chars input.
        ld      de,0
        ld      e,a             ;load this number into de,
        add     hl,de           ;and add to hl so we point to
        inc     hl              ;end of input.
        ld      a,'H'
        ld      (hl),a          ;set up string for ahex.
;
;Convert the input arguments into binary, and load into locations 6100h
;and 6200h for the 68000.
;
        ld      bc,stor1+1
        call    ahex            ;call ahex (in asmlib).
        ld      c,h
        ld      b,l             ;want to store hi byte first for 68000.
        ld      (6100h),bc
        ld      bc,stor2+1
        call    ahex            ;convert to binary.
        ld      c,h
        ld      b,l
        ld      (6200h),bc
;
;Zero out area into which answer is to be placed.
;
        ld      b,4
        xor     a
        ld      hl,1000h
nul:    ld      (hl),a
        inc     hl
        djnz    nul
;
;Save CDOS low memory and reinitialize addrs. 0h thru 7h for reset
;sequence of 68000 --  register a7 is loaded with four bytes beginning
;at 0h; the program counter is loaded with the next sequential four bytes.
;
        ld      de,700h         ;save CDOS low memory.
        ld      hl,0
        ld      bc,0ffh
        ldir                    ;move memory.
        ld      hl,0            ;init addrs. zero for 68000 stack.
        ld      a,0
        ld      b,6
loop:   ld      (hl),a
        inc     hl
        djnz    loop            ;init 03h for 68000 program counter.
        ld      a,60h
        ld      (hl),a
        inc     hl
        xor     a
```

```
        ld      (hl),a
;
;Turn on 68000 processor to do the multiply and place the result at
;1000h.  This is done by firmware on the DPU whenever the Z-80 does
;an I/O output to port 0FFh. It outputs 1 to turn on the DPU.
;
        ld      a,1
        out     0ffh,a          ;turn on 68000
        ld      de,0            ;if we get back, restore CDOS low memory
        ld      hl,700h         ;so we can do system calls.
        ld      bc,0ffh
        ldir
;
;Convert result of multiplication to ASCII and output it.
;
        ld      hl,string
        ld      a,(1000h)
        call    binh2           ;binary to ASCII conversion (in Asmlib).
        ld      hl,string+2
        ld      a,(1001h)
        call    binh2
        ld      hl,string+4
        ld      a,(1002h)
        call    binh2
        ld      hl,string+6
        ld      a,(1003h)
        call    binh2
        ld      de,answer
        ld      c,9
        call    5
        ld      de,700h         ;save CDOS low memory.
        ld      hl,0
        ld      bc,0ffh
        ldir                    ;move memory
        pop     bc
        dec     b
        jp      nz,go_on
        ld      de,contin
        call    prnbf$
        ld      c,80h           ;don't echo input.
        call    5               ;call CDOS
        cp      0dh
        jp      z,mainlp        ;hitting any key but <RET> will exit.
        jp      0
;
        list    text
;
msg:    db      0ah,0dh,0ah,0dh,'  This program prompts for two hex numbers'
        db      ' in the range 0h-FFFFh.',0ah,0dh
        db      '  Only the digits 0-9 and A-F (caps only) give proper'
        db      ' results.',0dh,0ah,'  The answer is displayed in hex.'
        db      0ah,0dh,'  Out-of-range input produces unexpected answers.'
        db      0dh,0ah,'  After any five multiplies, you may exit.'
```

```
        db      0dh,0ah,0dh,0ah,'$'
answer: db      0dh,0ah,0dh,0ah,'  Your answer, expressed'
        db      ' in hexadecimal is  >>  '
string: ds      8
        db      0dh,0ah,0dh,0ah,'$'
arg1ms: db      0dh,0ah,0dh,0ah,'  Input first'
        db      ' hex argument (digits 0-F)  >>  ','$'
arg2ms: db      0dh,0ah,'  Input second hex argument (digits 0-F) >>  ','$'
stor1:  ds      6
stor2:  ds      6
contin: db      0dh,0ah,0dh,0ah,'  To continue, hit <RETURN>. '
        db      ' Any other key exits.'
        db      0dh,0ah,0dh,0ah,'$'
        db      '$'

;The following 68000 program looks for values to multiply together in
;addresses 6100h and 6200h.  The values are placed there by
;the Z-80 processor. The result of the multiplication is placed in
;location 1000h to 1003h. These comments are what the 68000 assembly code
;would look like, but if you don't have the 68000 Assembler you can
;hand assemble the program as was done in this example.
;
;       org     6000h
;start: move    6100h,d0        ;get 16 bit value into d0.
;       mulu    6200h,d0        ;do unsigned multiply from memory
;                               ;to data register.
;       move.l  d0,1000h        ;store long word result back
;                               ;into memory.
;
;The next line moves the value at 6500h to 0FFFFh.  The value is 0,
;and was placed there by the Z-80.  The address 0FFFFh is sign extended
;by the 68000 and so represents absolute address FFFFFFFFh.  This resides
;in the top-most 64k bytes of the 68000 address space.  The DPU firm-
;ware is designed to interpret any address access in this top 64k
;space as an I/O access.  It is turned into an input or output to
;hardware ports -- in this case, port 0FFh.  In effect this outputs
;zero to port 0FFh.  Doing this turns off the 68000 processor and
;at the same time turns on the Z-80.  That is, we may return control
;to the Z-80 in this way.
;This technique for I/O is used because the 68000 is memory mapped.
;
;       move.b  6500h,0FFFFh    ;output contents of 6500h to port 0FFh.
;
;Next line will be executed the next time that the Z-80 turns
;on the 68000.  For this example, this will be an absolute
;jump to the beginning of this 68000 program.
;
;       jmp     6000h
;
;       end     start
;
;Stored at the label "prog68" is the machine code represented by
;the above program segment.  This code is moved by the Z-80 to
```

```
;address 6000h.  Before turning on the 68000, the value
;00006000h is placed at address 4h,5h,6h,7h.  After reset, the
;68000 will load its program counter from these addresses the first
;time it is turned on.  These addresses MUST BE initialized before
;transferring control to the 68000.  If not, whatever happens to be
;at 4h,5h,6h,and 7h will go into the 68000 program counter and
;the 68000 will begin executing where ever the pc is pointing.
;
prog68: db      30h,38h,61h,0,0c0h,0f8h,62h,0,23h,0c0h,0,0,10h,0
        db      11h,0f8h,65h,0,0ffh,0ffh,4eh,0f9h,0,0,60h,0
count:  dl      $-prog68
;
        end     start
```

Appendix B

LIMITED WARRANTY

Cromemco, Inc. ("Cromemco") warrants this product against defects in material and workmanship to the original purchaser for ninety (90) days from the date of purchase, subject to the following terms and conditions.

What Is Covered By This Warranty:

During the ninety (90) day warranty period Cromemco will, at its option, repair or replace this Cromemco product or repair or replace with new or used parts any parts or components, manufactured by Cromemco, which prove to be defective, provided the product is returned to an Authorized Cromemco Dealer as set forth below.

How To Obtain Warranty Service:

You should immediately notify IN WRITING your Authorized Cromemco Dealer or Cromemco of problems encountered during the warranty period. In order to obtain warranty service, first obtain a return authorization number by contacting the Authorized Cromemco Dealer from whom you purchased the product. Then attach to the product:

1. Your name, address and telephone number,
2. the return authorization number,
3. a description of the problem, and
4. proof of the date of retail purchase.

Ship or otherwise return the product, transportation and insurance costs prepaid, to the Authorized Cromemco Dealer. If you are unable to receive warranty repair from the Authorized Cromemco Dealer from whom you purchased the product, you should contact Cromemco Customer Support at: Cromemco, Inc., 280 Bernardo Ave., Mountain View, Ca. 94043.

What Is Not Covered By This Warranty:

Cromemco does not warrant any products, components or parts not manufactured by Cromemco.

This warranty does not apply if the product has been damaged by accident, abuse, misuse, modification or misapplication; by damage during shipment; or by improper service. This product is not warranted to operate satisfactorily with peripherals or products not manufactured by Cromemco. Transportation and insurance charges incurred in transporting the product to and from the Authorized Cromemco Dealer or Cromemco are not covered by this Warranty.

Exclusion of Liability, Damages, and Other Warranties:

THIS WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, WHETHER ORAL OR WRITTEN, EXPRESS OR IMPLIED. ANY IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF PURCHASE OF THIS PRODUCT. IF THIS PRODUCT IS NOT IN GOOD WORKING ORDER AS WARRANTED ABOVE, YOUR SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. CROMEMCO SHALL NOT BE LIABLE FOR INCIDENTAL AND/OR CONSEQUENTIAL DAMAGES FOR THE BREACH OF ANY EXPRESS OR IMPLIED WARRANTY, INCLUDING DAMAGE TO PROPERTY AND, TO THE EXTENT PERMITTED BY LAW, DAMAGES FOR PERSONAL INJURY, EVEN IF CROMEMCO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE AGENTS, DEALERS, AND EMPLOYEES OF CROMEMCO ARE NOT AUTHORIZED TO MAKE MODIFICATIONS TO THIS WARRANTY, OR ADDITIONAL WARRANTIES BINDING ON CROMEMCO ABOUT OR FOR PRODUCTS SOLD OR LICENSED BY CROMEMCO. ACCORDINGLY, ADDITIONAL STATEMENTS WHETHER ORAL OR WRITTEN EXCEPT SIGNED WRITTEN STATEMENTS FROM AN OFFICER OF CROMEMCO DO NOT CONSTITUTE WARRANTIES AND SHOULD NOT BE RELIED UPON. SOFTWARE, TECHNICAL INFORMATION AND FIRMWARE ARE LICENSED ONLY BY A SEPARATE AGREEMENT ON AN "AS IS" BASIS.

Limitation on Statute of Limitation and Transferability:

This warranty and the statute of limitations shall run concurrently with any acceptance period. This warranty is not transferable. No suit, litigation, or action shall be brought based on the alleged breach of this warranty or implied warranties more than one year after the date of purchase in those jurisdictions allowing such a limitation, otherwise no such action shall be brought more than one year after the expiration of this warranty.

Other Important Provisions:

Some states do not allow the exclusion or limitation of incidental or consequential damages or limitations on how long an implied warranty lasts, so the above limitation or exclusion may not apply to you. This warranty shall not be applicable to the extent that any provision of this warranty is prohibited by any federal, state or municipal law which cannot be preempted. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

7/82